IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

INVENTORS:	Stephan L. JOURDAN, et al.
SERIAL NO:	09/608,624
FILING DATE:	June 30, 2000
TITLE:	TRACE INDEXING VIA TRACE END ADDRESSES
ART UNIT:	2183
EXAMINER:	Henry TSAI

Mail Stop Appeal Brief - Patents
COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, VA 22313-1450

SUPPLEMENTAL APPEAL BRIEF

SIR:

This supplemental brief is in response to the Office Action mailed April 5, 2006, reopening prosecution in the above-identified application. The Appellant requests reinstatement of the appeal.

Response to new grounds of rejection

The above-identified Office Action (hereafter, "Office Action") raised new grounds of rejection under 35 USC § 101, 35 USC § 112, 1st and 2nd paragraphs, and 35 USC § 102. The Appellant's response follows.

Pursuant to 37 CFR § 1.116, cancellation of claims 28-30 without prejudice is requested. Claims 1-7, 9-20, 23-27 and 38-43 remain pending in the application, and all stand rejected.

Section 101

Claims 1-3, 9-15, 20, 23-30 and 38-43 were rejected under 35 USC § 101 as being directed to non-statutory subject matter. Specifically, the Examiner contends that the language "a memory entry storing a trace" as recited in claim 1, for example, is "descriptive material per se and is non-statutory because it is not capable of, by itself, causing functional change in the computer." Office Action at page 3, 3rd paragraph.

The § 101 rejection is error. First, removing any possible ambiguity, claim 1 recites an "[a]pparatus comprising a memory entry storing a trace ... . " Thus, claim 1 recites at least a machine or an article of manufacture and is therefore statutory.

Further, the Examiner errs in stating that a memory entry storing a trace cannot cause a functional change in a computer. To the contrary, a trace is a sequence of computer instructions; changing the state of a computer is exactly what computer instructions do.

Independent claim 20 recites an "[a]pparatus, comprising a memory entry storing a sequence of program instructions as a trace ... ." Thus, the preceding remarks are equally applicable to claim 20.

Independent claim 23 recites a "memory comprising storage for a plurality of traces ... ." The person of ordinary skill in the computer-related arts understands a memory to be a component of a computer for electronically storing data. Many dictionaries recognize this meaning; one example is the popular online dictionary www.dictionary.com, which defines a memory as "a unit of a computer that preserves data for retrieval." A trace, as noted earlier, is a sequence of computer instructions that change the state of a computer.

Independent claim 38 recites an "[a]pparatus, comprising: a memory ... and ... a trace ... ." Similarly, independent claims 41-43 each recites an apparatus comprising a memory or memory entry storing a trace, or a memory storing a trace.

In view of the above, each of independent claims 1, 20, 23, 38 and 41-43 recites at least a machine or article of manufacture storing a sequence of instructions to change a state of a computer, and therefore recites statutory subject matter under § 101.

The Examiner rejects claim 9 as relating to "just an abstract idea." The Examiner argues: "The claim does not provide practical application that produces a useful, tangible and concrete result. Therefore, this claim is non-statutory." Office Action at page 4, 2nd paragraph.

This is error. As stated by the Federal Circuit,

> every step-by-step process, be it electronic or chemical or mechanical, involves an algorithm in the broad sense of the term. Since § 101 expressly includes processes as a category of inventions which may be patented and § 100(b) further defines the word "process" as meaning "process, art or method, and includes a new use of a known process, machine, manufacture, composition of matter, or material," it follows that it is no ground for holding a claim is directed to nonstatutory subject matter to say it includes or is directed to an algorithm. This is why the proscription against patenting has been limited to mathematical algorithms . . . .

In re Iwahashi, 888 F.2d 1370, 1374, 12 USPQ2d 1908, 1911 (Fed. Cir. 1989) (emphasis in the original). Claim 9 clearly does not set forth a mathematical algorithm. Instead, it sets forth a step-by-step process that does not fall within the "abstract idea" exception to patentable subject matter.

Claim objections

Claim 15 was objected to as being informal. This is error. Claim 15 refers back to the "selected block" of claim 9. There can be no confusion about which block is meant, since only one block is associated with a selecting step in claim 9.

Claim 39 was objected to as being of improper dependent form for failing to further limit the subject matter of a previous claim. This is error. Claim 38 recites "a trace having a multiple-entry, single exit architecture." Claim 39 further limits the subject matter of claim 38 by specifying that the trace is a "complex" trace and that it has "multiple independent prefixes and common, shared suffix." This information imparts further specificity and structure to the trace recited in claim 38.

Section 112

1st paragraph

Claims 23-27 and 41-43 were rejected under 35 USC 112, 1st paragraph as failing to comply with the written description requirement. Specifically, the Examiner alleges that claims 23 and 41-43 recite indexing means but that the specification contains no supporting description. Office Action, page 7, 2nd paragraph.

This is error. The specification has ample support for the rejected claims. For example, the specification in the paragraph bridging pages 2 and 3 reads, "Embodiments of the present invention assemble a new type of traces [sic], called 'extended blocks' herein, according to an architecture that permits several entry points but only a single exit point. These extended blocks *may be indexed based upon the address of the last instruction therein* " (emphasis added). The last sentence of the 3rd paragraph of page 3 reads, "Again, the block cache 280 may *index the extended blocks based upon an IP of the terminal instruction in the block*" (emphasis added). Claim 3 as originally filed reads, "The trace of claim 1, wherein the trace is *indexed by an address of a terminal instruction therein*" (emphasis added).

2nd paragraph

Claims 1-3, 4-7, 20, 23-30 and 41-43 were rejected under 35 USC 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which the applicant regards as the invention. This is error.

Regarding claim 1 in particular, the Examiner contends as follows: "an 'entry' is ... nothing more than *a unit of information*. Thus, it is not clear how claim 1 constitutes an apparatus and how a memory entry can store anything" (emphasis in original). Office Action, the paragraph bridging pages 7 and 8.

In response, it is observed that claim 1 recites not merely an "entry" but a "memory entry." As discussed previously, a memory is a unit of a computer that preserves data for retrieval. A memory entry, thus, is a sub-unit of a computer able to store at least a subset of the data that the memory as a whole can store.

Further concerning claim 1, the Examiner alleges that (1) "it is not clear how a 'trace' can have a multi-entry and a single exit" and that (2) "[t]he flow of a program can only have one entry in a trace." Office Action, page 8, 2nd paragraph. In response to (1), it is observed that the Examiner has applied a § 112, 2nd paragraph rejection here. It is not the role of the claims, under § 112, 2nd paragraph, to explain or describe the invention, but to "particularly point out and distinctly claim" the invention. Claim 1 does so.

The meaning of (2) is obscure. One possible interpretation is "the flow of a program can have entries, but only one of them can be in a trace." If this is what is meant, the Appellant's response is that it does not appear to correspond in an understandable way with anything in the present specification or claims.

According to another possible interpretation, (2) would seem to subsume a program within a trace. In other words, another possible interpretation is "when a program is in a trace, the flow of the program can have only one entry point." The latter, again, does not appear relevant to the present specification or claims, which relate to a trace architecture with multiple entries and a single exit. If the Examiner is attempting to argue that no such trace architecture can exist, the Appellant's response is that the Examiner is mistaken: the present application discloses and claims just such a trace architecture.

In support of (2), the Examiner argues that (3) "[t]he flow of a program can only have one entry in a trace ... since the trace is dependent on the input to the program." Office Action, page 8, 2nd paragraph. As a basis for the foregoing, the Examiner cites "Nair Ravi, 'Trace Caches and Trace Processors', lecture note, Pages 1-2, USPTO training for examiners in the computer architecture area, 4/29/04)" (hereafter, "Ravi"). Id.

The cited portion of Ravi discloses that a "trace of a program is the exact sequence of instructions executed by the computer while running the program," and that "[f]or a given program, the trace is dependent on the input to the program." Ravi, page 1. The latter is repeated, as noted above, by the Examiner in (3).

The Appellant responds that in (3) the Examiner juxtaposes two entirely unrelated ideas. Ravi's "the trace is dependent on the input to the program" means nothing more than what it plainly says: that a trace depends on a program's input. This places no restrictions whatsoever on, and in fact has nothing at all to do with, how a trace is architected in terms of entries and exits.

Concerning claim 3, the Examiner states,

> "In claim 3, it is not clear what is meant by 'terminal instruction therein'. It appears that the terminal instruction is inside a trace stored in the memory entry. However, a trace comprises the sequences of executed instructions (see page 1, lines 16 in the specification), e.g., it comprises the address data (or pointers) of executed instructions. The address data is not an instruction. Therefore, a trace does not have terminal instruction therein."

Office Action, paragraph bridging pages 8 and 9. To point out only the most obvious error and self-contradiction in this paragraph: the Examiner observes that a trace comprises a sequence of executed instructions, and then concludes that it does not have a terminal instruction. This is impossible.

Concerning claim 4, the Examiner states, "it is not clear how to define 'complex blocks' and 'block prefixes' since the structural relationship between 'complex blocks', 'block prefixes', and the front-end system was not defined." Office Action, page 9, 2nd paragraph. Here, again the Examiner seeks to force the role of the written description onto the claims, apparently taking the position that terms whose meanings are self-evident need to be bolstered with added description. As discussed previously, this is not required of the claims. Notwithstanding -- a "block prefix" is simply that – a prefix of a block. The prefixes are for blocks labeled "complex." The relationship between the complex blocks, their prefixes and the front-end system is that the complex blocks and block prefixes are elements of the front-end system, which is perfectly clearly from the fact that the language reciting the complex blocks and block prefixes follows "A front-end system for a processor, comprising:".

As to claim 6, the Examiner alleges that "it is not clear what is meant by 'blocks having a multiple-entry, single exit architecture'. Some essential elements or more detailed descriptions are missing." Office Action, page 9, 4^3rd paragraph. The Appellant again points out that "detailed descriptions" are not a statutory requirement for the claims.

As to claim 20, the Examiner states that "it is not clear what is meant by 'a last instruction in the memory entry'," and repeats the self-contradictory argument made concerning claim 3, i.e., that a sequence of instructions does not have a last instruction. See Office Action, paragraph bridging pages 9 and 10. The Appellant reiterates that this is error.

Further concerning claim 20, the Examiner states that "it is not clear how a trace has multiple separate prefixes." Office Action, page 10, 2nd paragraph. The Appellant repeats that the claims are not required to explain the invention. In support of the proposition that "it is not clear how a trace has multiple separate prefixes," the Examiner repeats the contention, made earlier in connection with claim 1, that "[t]he flow of a program can only have one entry in a trace," and again cites the Ravi reference. Id. As these issues were addressed by the Appellant earlier in the discussion of the § 112, 2nd paragraph rejection of claim 1, the Appellant refers the reader to these earlier remarks.

Regarding claim 23, the Examiner states that "it is not clear how to define 'a last instruction therein' since there may have [sic] many last instructions in the memory." Office Action, page 10, 3rd paragraph. In claim 23, "a last instruction therein" refers to a last instruction of "traces," not of a memory.

Regarding claim 27, the Examiner states that "it is not clear how a trace can include executable instructions. As set forth above, a trace comprises the sequence of executed instructions." Office Action, paragraph bridging pages 10 and 11. As others of the Examiner's statements have been, the latter statements are self-contradictory: in the same breath, the Examiner says that a trace doesn't include instructions, and then that it does. This is error.

Concerning claims 23 and 41-43, the Examiner states that these claims "recited indexing means, however, in the specification there's no description about the structure for the indexing means necessary for supporting the claims." Office Action, page 11, 2nd paragraph. In response, the Appellant observes that the Examiner has applied a § 112, 2nd paragraph rejection here. The Examiner's comments, however, appear to relate to the requirements of § 112, 1st paragraph. In any event, ample support for indexing means has been previously demonstrated earlier.

Section 102

Claims 1-3 and 38-40 were rejected under 35 USC § 102(b) as being anticipated by Kaylor (U.S. 5,492,276) ("Kaylor"). This rejection constitutes unmitigated error. Not only does the Kaylor reference not disclose the elements of the rejected claims as required under § 102, it is not even remotely related to the same field as the present invention. The present invention, as disclosed and claimed, relates to computers, and in particular to processor front-ends and techniques for managing them. The Kaylor reference, by contrast, relates to plumbing. Not a single element in the Kaylor reference can fairly be found under any reasonable interpretation to correspond to any element of the rejected claims.

Claims 1, 2, 16, 20, 28, 29, 38 and 39 were rejected under 35 USC § 102(e) as being anticipated by Agarwal (US 5,966,541). Here, the Examiner repeats the arguments of the final rejection mailed January 4, 2005, which were addressed in the appeal brief filed June 28, 2005. Accordingly, the Appellant's arguments will not be repeated here.

Conclusion

In view of the above, it is clear that the Examiner erred in the rejections and objections asserted in the Office Action. It is therefore respectfully requested that the Board reverse the Examiner, withdraw all rejections and objections, and allow claims 1-7, 9-20, 23-27 and 38-43.

The Examiner is invited to contact the undersigned at (202) 220-4323 to discuss any matter concerning this application. The Office is authorized to charge any fees related to this communication to Deposit Account No. 11-0600.

Respectfully submitted,

Dated: July 21, 2006

By:
William E. Curry
Reg. No. 43,572

KENYON & KENYON LLP
Attorneys for Intel Corporation
1500 K Street, N.W., Suite 700
Washington, D.C. 20005
Tel: (202) 220-4200
Fax:(202) 220-4201